IN THE SPECIFICATION

Please replace paragraph 37, with the following rewritten paragraph:

Figures 18A through ~~18D~~ 18H show a Processor Builder class according to one embodiment of the present invention;

Please replace paragraph 42, with the following rewritten paragraph:

Figures 23A-23D ~~and 23B~~ are class specifications of the CFormatProtocol_InformationBase class, that interfaces with the System Manager computer code device;

Please replace paragraph 44, with the following rewritten paragraph:

Figures 25A[[,]]-25E ~~25B and 25C~~ are class specifications of the CProtocolRestrictionCheck class, where the steps in the oneFormalRestriction function shows the process to modify the map structure.

Please replace paragraph 108, with the following rewritten paragraph:

Figures 18A-18H show ~~Figure 18A shows~~ the function list and the attributes of the CProcessorBuilder Class according to one embodiment of the present invention. The public function createDataFormatProcessor receives the specification for the Data Formatter and returns the pointer to the specified Data Formatter object in the abstract class type. The public function createProtocolProcessor function receives the specification for the Protocol Processor and returns the pointer to the specified Protocol Processor in the abstract class type. The m_pDataFormatter attribute of the class is used to cache the specified data formatter in the class. The other two map attributes show the structure shown in Figures 15 and 17. The

function definition section shows the steps used by the various functions declared in the

function list.

Please replace paragraph 113, with the following rewritten paragraph:

Figures 23A-23D illustrate ~~Figure 23A~~ is an exemplary class definition of

CFormatProtocol_InformationBase interface/class. The functions, storeFormatAndProtocol

and GetFormatAndProtocolVector, are public functions used by the System Manager 560

computer code device (an object of CMonitorManager class). Two map structures keep the

specified formats and protocols passed through the function, storeFormatAndProtocol, after

checking the validity of the combination of the format and the protocol through the object,

m_FormatProtocolCombinationCheck of CFormatProtocolCombinationCheck class. The

class also contains the object, m_ProtocolRestrictionCheck of CProtocolRestrictionCheck

class. The flag, m_bFirstGetCall is used to call the function in the

m_ProtocolRestrictionCheck when the function getFormatAndProtocolVector is called for

the first time. The attribute, m_FormatProtocolVectorMapIterator, is used by the

getFormatAndProtocolVector function to iterate over the m_FormatProtocolVectorMap.

Figures 23C and 23D shows the steps for three main functions in the

CFormatProtocol_InformationBase class 600.

Please replace paragraph 115, with the following rewritten paragraph:

Figures 25A, 25B, and 25C show the class definition of the

CProtocolRestrictionCheck class 620. The m_bOneFormatRestriction attribute specifies

whether the information is restricted to a single format. Other types of restrictions could be

implemented by adding other private functions and attributes. An exemplary restriction

algorithm for the present invention is illustrated in Figures ~~25B and 25C~~ 25D and 25E

(showing the steps in the private function oneFormatRestriction).